实验9 实现访存指令的CPU设计实验

一、实验目的

- 掌握RISC-V I型取数指令和S型存数指令的数据通路设计；
- 掌握指令流和数据流控制的方法；
- 学习依据新增指令，修改多周期CPU系统结构的方法
- 具备连接各模块构建整机的能力
- 实现I型取数指令和S型存数指令的功能

二、实验内容与原理

- **实验内容：在构建并实现了运算类指令（R型和I型）、传送指令（U型lui指令）的RISC-V CPU基础上，新增两条访存指令lw和sw**
- **构建数据存储器与MDR寄存器，并添加到上次实验RISC-V CPU中，与各个逻辑模块正确连接**
- **修改二级译码及控制单元，实现新的目标指令集的功能**
- **编写测试程序，检验CPU是否能正确执行基于新目标指令集的程序**

1. 新增目标指令
2. 访存指令的数据通路与执行过程
3. 控制单元CU
4. 测试程序

1、新增目标指令

I型取数指令lw编码及功能

位数	12位	5位	3位	5位	7位	指令功能
I型指令字段	imm12	rs1	funct3	rd	opcode	
汇编指令	机器指令编码					
lw rd,imm12(rs1)	imm[11:0]	rs1	010	rd	0000011	Mem32[rs1+SE32(imm12)]→rd

- lw指令执行存储器的读操作

1、新增目标指令

S型存数指令sw编码及功能

位数	7位	5位	5位	3位	5位	7位	指令功能
S型指令字段	imm7	rs2	rs1	funct3	imm5	opcode	
汇编指令	机器指令编码						
sw rs2,imm12(rs1)	imm[11:5]	rs2	rs1	010	imm[4:0]	0100011	rs2→Mem32 [rs1+SE32(imm12)]

- sw指令执行存储器的写操作

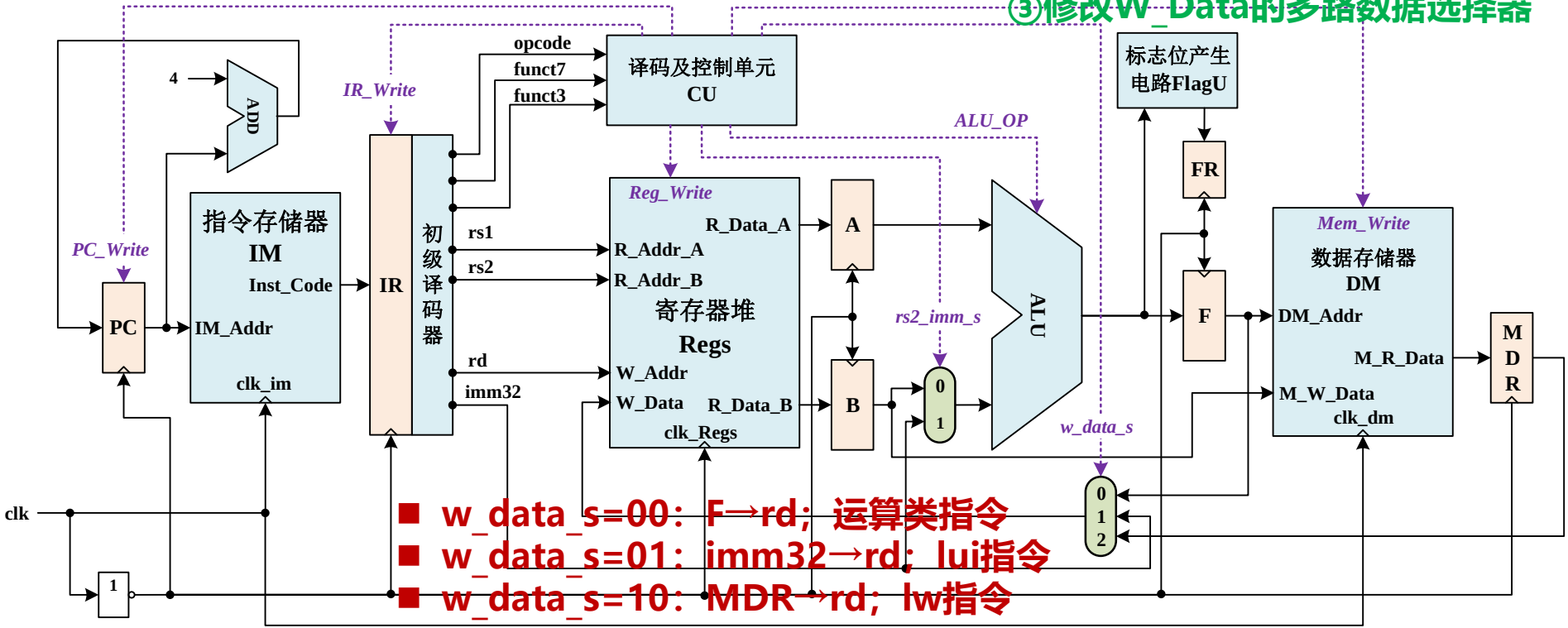
2、访存指令的数据通路与执行过程

- RISC CPU：只有访存指令才访问存储器，且不对存储器操作数进行运算
- 需要实现数据存储器，三个问题：
  - ①访问存储器的地址从哪里来?
    - EA=rs1+SE32(imm12)，ALU完成加法，结果在F中
    - 访问存储器的地址可以直接由F提供
  - ②读出操作时，读出的数据到哪里去?
    - 读出的数据可以保存到MDR寄存器
    - 写回周期：MDR→rd，W_Data的二选一数据选择器要变成三选一
  - ③写入操作时，写入的数据由谁提供?
    - sw指令执行存储器的写入操作，写入的数据是rs2寄存器的内容
    - rs2寄存器内容：保存在B暂存器中的
    - 写入操作时，写入的数据由暂存器B提供

添加访存指令之后的CPU系统结构图

相比实验8，主要添加或修改了三个部件：

①添加一个数据存储器
②添加暂存器MDR
③修改W_Data的多路数据选择器



- w_data_s=00：F→rd；运算类指令
- w_data_s=01：imm32→rd；lui指令
- w_data_s=10：MDR→rd；lw指令

指令的执行流程图

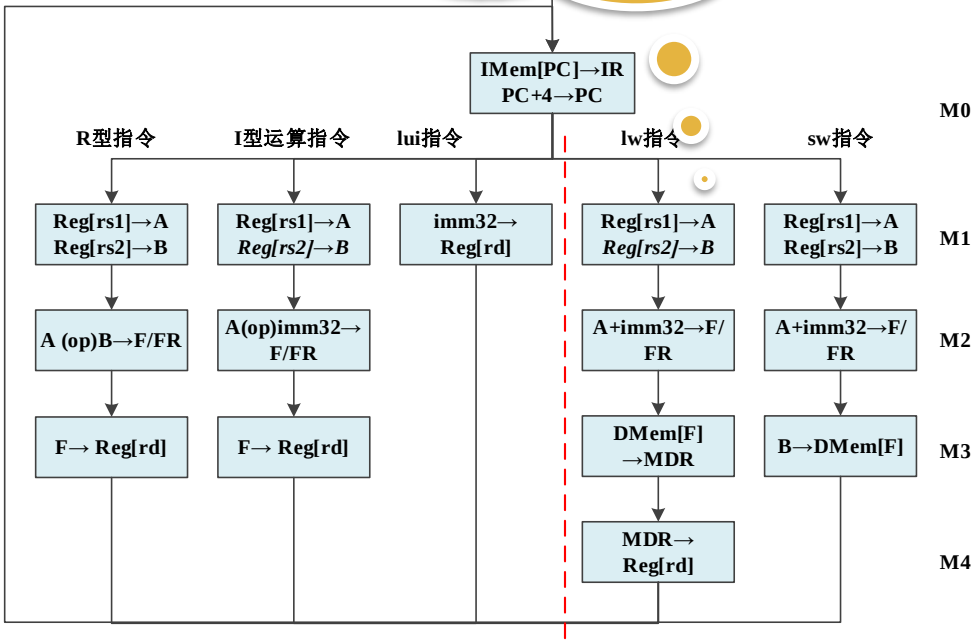
22条指令，包括实验8的运算和传送指令，新指令lw和sw

指令周期最长的指令

- lw指令
- M1周期：读出rs1内容到A；
- M2周期：执行有效地址计算，存入暂存器F；
- M3周期：执行存储器读操作，读出数据暂存MDR；
- M4周期：MDR写回目的寄存器rd；

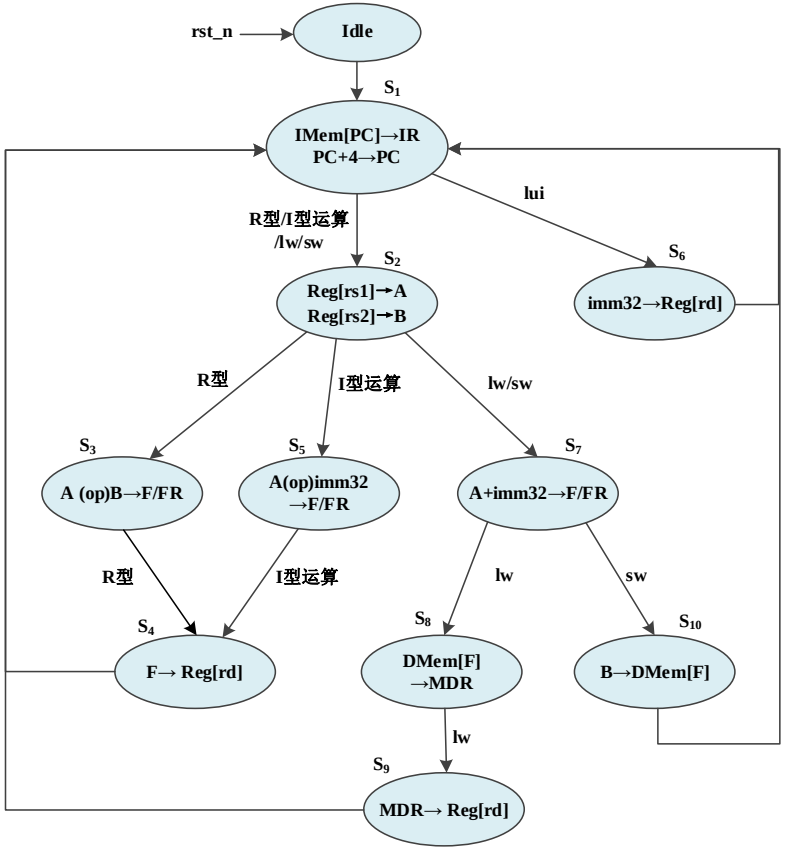
- sw指令
- M1周期：读出rs1内容到A、rs2内容到B；
- M2周期：执行有效地址计算，存入暂存器F；
- M3周期：将B写入存储器中；

每个clk周期，MDR都会打入存储器读出的数据吗？MDR的数据正确吗？对别的部件有影响吗？



3、控制单元CU

- （1） 指令执行的状态转移图
- CU的状态转移图多了4个状态S7~S10
  - S7状态计算有效地址EA；
  - S8执行存储器读操作；
  - S9执行写回操作；
  - S10执行存储器的写操作。



lw指令的状态转移流程是：

S1→S2→S7→S8→S9→S1

sw指令的状态转移流程是：

S1→S2→S7→S10→S1

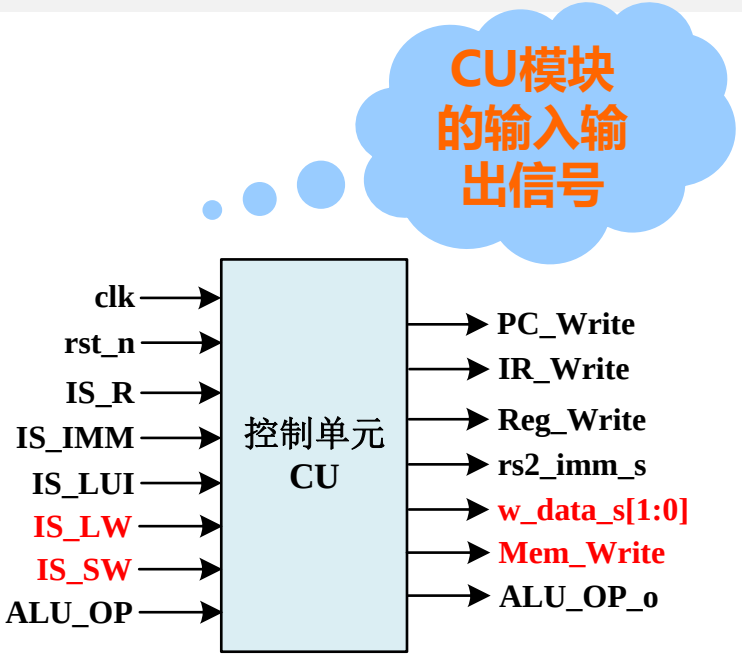
（2）状态输出的控制信号

状态 / 控制信号	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
PC_Write	1	0	0	0	0	0	0	0	0	0
IR_Write	1	0	0	0	0	0	0	0	0	0
Reg_Write	0	0	0	1	0	1	0	0	1	0
Mem_Write	0	0	0	0	0	0	0	0	0	1
ALU_OP[3:0]	—	—	****	—	****	—	0000	—	—	—
rs2_imm_s	—	—	0	—	1	—	1	—	—	—
w_data_s[1:0]	—	—	—	00	—	01	—	—	10	—

（2）状态输出的控制信号

- 修改有限状态机的状态输出函数，修改发送的控制信号，就实现了能控制22条指令执行过程的控制单元CU。

- 输入信号多了两条指令译码信号——IS_LW、IS_SW，是由二级译码模块产生的，故还需要修改二级译码模块ID2；输出信号多了Mem_Write，而w_data_s变成2位。



- 数据存储器在clk的上跳沿执行写操作，此时要检测到Mem_Write=1，才会执行写入操作；因此，需要提前赋值（即S7状态就赋值）：

```
assign Mem_Write = (Next_ST==S10);
```

4、测试程序范例


```
main:
 addi x31,x0,16        #x31=16
 lw     x26,0(x31)     #x26=DMem[16]
 lw     x27,4(x31)     #x27=DMem[20]
 sw     x27,0(x31)     #DMem[16]=x27
 sw     x26,4(x31)     #DMem[20]=x26
 lw     x26,0(x31)     #x26=DMem[16]
 lw     x27,4(x31)     #x27=DMem[20]
 add  x28,x26,x27   #x28= x26+ x27
 sw   x28,16(x0)     # DMem[16]=x28
 lw    x29,16(x0)     #x29=DMem[16]
 addi x1,x0,-0x78A #x1=0xFFFF_F876
 addi x2,x0,4          #x2=0x0000_0004
 add  x3,x1,x2         #x3=0xFFFF_F87A
 sub  x4,x1,x2         #x4=0xFFFF_F872
 sll    x5,x1,x2       #x5=0xFFFF_8760
 srl    x6,x1,x2       #x6=0x0FFF_FF87
 sra   x7,x1,x2        #x7=0xFFFF_FF87
 slt   x8,x1,x2    #x8=0x0000_0001
```

```
sltu  x9,x1,x2          #x9=0x0000_0000
and  x10,x5,x6          #x10=0x0FFF_8700
or     x11,x5,x6        #x11=0xFFFF_FFE7
xor   x12,x5,x6         #x12=0xF000_78E7
lui    x13,0x80000      #x13=0x8000_0000
addi  x14,x13,-1        #x14=0x7FFF_FFFF
addi  x15,x14,0x123      #x15=0x8000_0122
slli    x16,x15,3        #x16=0x0000_0910
srli    x17,x15,3        #x17=0x1000_0024
srai    x18,x15,3        #x18=0xF000_0024
slti    x19,x18,-1       #x19=0x0000_0001
sltiu   x20,x18,-1       #x20=0x0000_0001
slti    x21,x18,1        #x21=0x0000_0001
sltiu   x22,x18,1        #x22=0x0000_0000
andi    x23,x12,0xFF     #x23=0x0000_00E7
ori     x23,x12,0xFF     #x23=0xF000_78FF
lui     x24,0x00010      #x24=0x0001_0000
addi    x24,x24,-1       #x24=0x0000_FFFF
xori    x25,x24,-1       #x25=0xFFFF_0000
```

- 10条lw和sw指令+实验8的测试程序
- 前5条指令：交换DM地址为16的单元(第5个字)和地址为20的单元(第6个字)的内容；
- 后5条指令：读出16号和20号单元内容，相加，结果存入16号单元
- 测试程序的机器指令代码：填入COE文件，用来初始化指令存储器
- 数据存储器：通过COE文件初始化数据

三、实验要求

1. 编写用于测试的汇编程序RIUS_test.s，可以自行设计，也可以采用前述范例程序；将其翻译成机器指令代码，写入RIUS_test.coe文件备用。
2. 将实验8工程另存为一个新工程，用RIUS_test.coe重新生成指令存储器模块。
3. 使用IP核新建一个数据存储器模块（64×32位），用COE文件初始化其内容，最好每个单元内容各不相同；新建一个MDR寄存器模块，改造W_Data的多路数据选择器为三选一数据选择器；修改二级指令译码模块ID2和控制单元CU；将它们与其他模块进行正确的连接，构建RIUS_CPU模块。
4. 对RIUS_CPU模块进行仿真，在rst_n之后，每来一个clk，观察指令执行的每一步骤（状态）是否符合预期，确保RIUS_CPU能正确执行目标指令集上的测试程序。

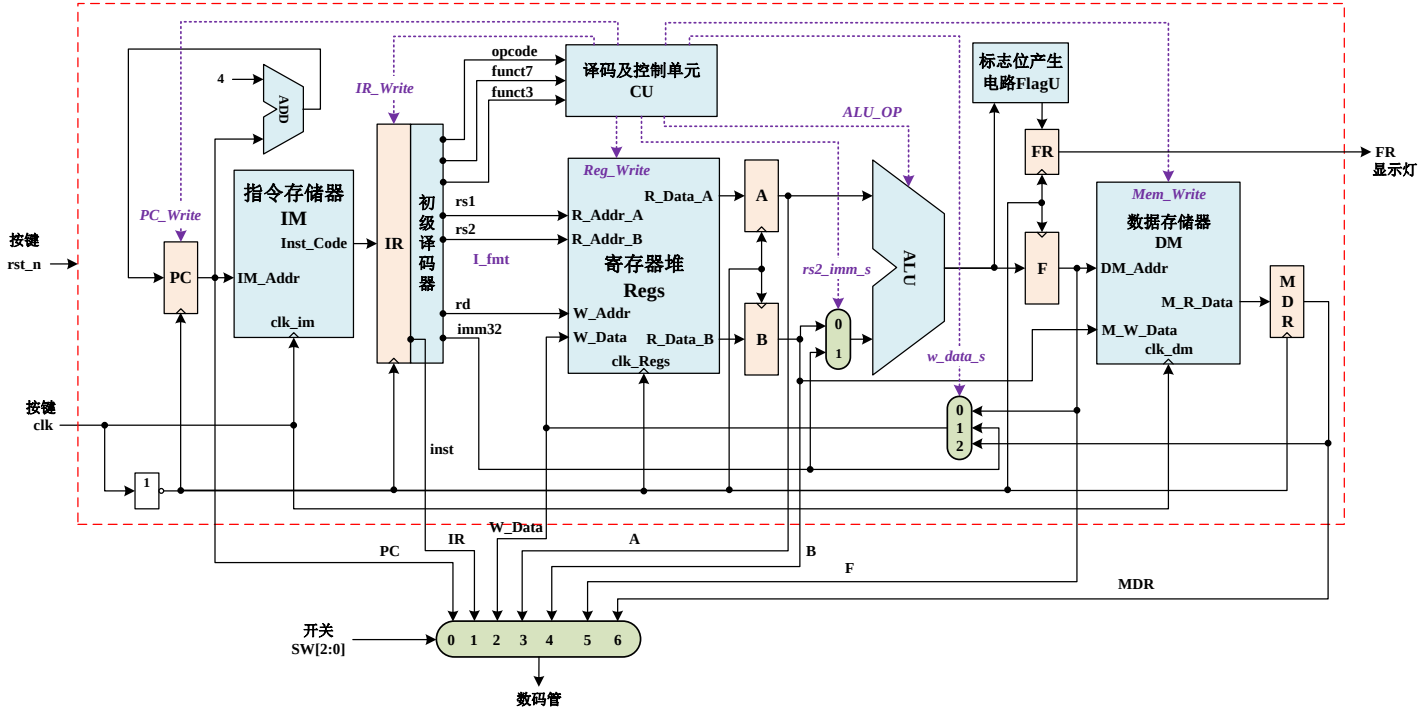
三、实验要求

5. 针对使用的实验板卡，设计RIUS_CPU的板级验证实验方案，编写顶层测试模块，要求至少能观察PC、IR、W_Data、MDR、标志位。同样需要修改RIUS_CPU模块的输出端口，将内部模块的相应信号引出。

- 一个板级验证实验方案：
  - 将PC、IR、W_Data以及暂存器A、B、F、MDR的内容都引出，经过一个7选1的数据选择器送到数码管显示，选择信号由三个开关
  - FR的四个标志位送到LED灯显示

三、实验要求

一个板级验证实验方案：



三、实验要求

6. 复位后，按clk按键，会按步骤执行指令，验证每条指令需要几个周期执行完，执行结果是否正确，将实验结果记录下来。

序号	PC	IR	汇编指令	预期执行结果	指令clk数	W_Data	MDR

三、实验要求

7. 撰写实验报告，重点内容包括：

1）对仿真结果进行分析；描述你设计的板级验证实验方案、模块结构与连接；说明你的板级操作过程；记录板级实验结果。

2）针对以下有待验证的实验问题，**给出证据与分析，得到有效结论**：

①lw指令取出的内存数据与数据存储器初始化值一致；

②sw指令能将数据存入指定内存地址；

③sw指令存入某单元的数据，通过lw指令读出，数据保持一致。

3）请力所能及回答或实践本实验的“**思考与探索**”部分。

四、实验步骤

1. 编写用于测试的汇编程序RIUS_test.s，可以自行设计，也可以采用前述范例程序；
2. 使用汇编器和反汇编器，将其翻译成机器指令代码，写入RIUS_test.coe文件备用；
3. 新建数据存储器的COE文件RIUS_data.coe备用，最好每个单元内容各不相同；
4. 将实验8的工程另存为一个新工程；
5. 用RIUS_test.coe重新初始化指令存储器的IP核模块IM；
6. 使用IP核新建一个64×32位的Single Port RAM，用RIUS_data.coe文件初始化其内容；

四、实验步骤

7. 新建一个数据存储器模块DM，调用上述IP核；
8. 新建MDR寄存器，与存储器DM正确连接；
9. 改造W_Data的多路数据选择器为三选一数据选择器；
10. 修改二级指令译码模块ID2，添加对lw和sw指令的译码；
11. 修改控制单元CU的有限状态机，添加lw和sw指令的状态及其控制；
12. 将DM、MDR、ID2、CU与其他模块进行正确的连接，构建RIUS_CPU模块；
13. 编写激励仿真代码，对RIUS_CPU模块进行仿真测试，首先rst_n=0，然后置1，之后，周期性地产生clk，观察输出信号，分析指令执行的每一步骤（状态）是否符合预期

四、实验步骤

14.设计RIUS_CPU模块的板级验证实验方案，然后据此编写一个顶层测试模块。按照要求至少能观察PC、IR、MDR以及写入寄存器堆的值W_Data；

15.新建管脚约束文件，进行相应的引脚配置；

16.生成*.bit文件，下载到实验设备的FPGA芯片中；

17.板级实验：按照你所设计的实验方案，操作输入设备、观察输出设备，预期的验证操作如下：

1）按rst_n按键，将PC和IR清零，接下来将从0号单元开始执行指令；

2）按动时钟键clk，每按一下，指令就执行一步，观察PC和IR的值，判断取数指令和存数指令是否能完成规定的功能；观察各部件值，判断其他各条指令执行结果是否正确。

五、思考与探索（至少完成1道）

1. 你的CPU在执行lw指令和sw指令时，是否按照你的程序，读出或者写入了指定字单元的数据？如果不是，请分析原因，更正CPU设计。
2. 你的测试程序，能否证明你的lw指令和sw指令是正确的？举例说明。
3. 本实验实现的CPU架构是哈佛结构还是普林斯顿结构？本实验的数据存储器和指令存储器地址空间重复，均是0~255（字节地址），如果将数据存储器的地址空间变为0x1000 0000~0x1000 00FF，请问要如何修改？
4. 如果实验5中已经实现了按照字节访问、按照半字访问的数据存储器，则本次实验可以尝试实现另外的访存指令：lb、lh、lbu、lhu、sb、sh指令。
5. 说说你在实验中碰到了哪些问题，你是如何解决的？